PhysioEx: a new Python library for Explainable Sleep Staging through Deep Learning

Guido Gagliardi[*,1,2,3], **Antonio Luca Alfeo**[1,4], **Mario G.C.A. Cimino**[1,4], **Gaetano Valenza**[1,4], **and Maarten De Vos**[2,5]

[1] Department of Information Engineering, University of Pisa, Pisa, Italy.

[2] Dept. of Electrical Engineering, KU Leuven, Belgium

[3] Dept. of Information Engineering, University of Florence, Italy

[4] Bioengineering & Robotics Research Center E. Piaggio, School of Engineering, University of Pisa, Pisa, Italy.

[5] Dept. of Development and Regeneration, KU Leuven, Belgium

∗ Correspondence to: `guido.gagliardi@phd.unipi.it`

Abstract.

Objective: Sleep staging is a crucial task in clinical and research contexts for diagnosing and understanding sleep disorders. This work introduces PhysioEx (Physiological Signal Explainer), a Python library designed to support the analysis of sleep stages using deep learning and Explainable AI (XAI).

Approach: PhysioEx provides an extensible and modular API for standardizing and automating the sleep staging pipeline, covering data preprocessing, model training, testing, fine-tuning, and explainability. It supports both low-resource devices and high-performance computing clusters and includes pretrained models based on the Sleep Heart Health Study (SHHS) dataset. These models support single-channel EEG and multichannel EEG-EOG-EMG configurations and are easily adaptable to custom datasets. PhysioEx also features a command-line interface toolbox allowing users to streamline the model development and deployment. The library offers a range of XAI post-hoc methods to explain model decisions and align them with expert knowledge.

Main results: PhysioEx benchmark state-of-the-art sleep staging models in a standard pipeline. Enabling a fair comparison between them both on the training source and out-of-domain sources. Its XAI techniques provide insights into deep learning-based sleep staging by linking model decisions to human-understandable concepts, such as AASM-defined rules.

Significance: PhysioEx addresses the need for a standardized and accessible platform for sleep staging analysis, combining deep learning and XAI. By supporting modular workflows and explainable insights, it bridges the gap between machine learning models and clinical expertise. PhysioEx is publicly available and installable via pip‡, making it a valuable tool for researchers and practitioners in sleep medicine.

Keywords: EEG, Sleep Staging, Explainable Artificial Intelligence, Deep Learning

‡ `https://pypi.org/project/physioex/`

1. Introduction

Sleep makes up about one-third of life, and good sleep is essential for physical and mental well-being[1]. Sleep disorders are linked to various health issues, from obstructive sleep apnea to narcolepsy. Diagnosing these involves assigning 30-second epochs of overnight polysomnography (PSG) data to sleep stages[2], a process called sleep staging. This is crucial for assessing sleep structure, such as cycles and duration of sleep bouts, aiding in the diagnostic process.

Manual approaches to sleep staging, outlined by the American Academy of Sleep Medicine (AASM), are conducted by clinicians in sleep labs. These approaches are time-consuming and resource-intensive, making them impractical for processing large datasets.

Significant advancements in automated sleep staging have been made[3, 4, 5, 6], particularly with the availability of large public datasets, like those from the National Sleep Research Resource (NSRR)[7]. By employing these datasets, Deep Learning (DL) architectures have been exploited, enabling machine learning algorithms to perform sleep staging with accuracy surpassing expert agreement levels, reaching clinical-grade performance[2].

Despite the promising performance of DL models, their deployment in clinical practice remains limited due to a lack of trust of clinicians[8]. Key reasons for this trust issue are: (i) the fact that DL models performances are affected by the randomness introduced by seeds and by different datasets used at training time, or even different train/test split of the same dataset, often resulting in unfair comparisons between different models, and hence difficulties to establish the best architecture to be employed in clinical practice; (ii) the "black-box" nature of AI systems, where the decision-making process is opaque, even to developers, leading to skepticism about the reliability of their results as well as the human acceptability of their decision-making process[9].

PhysioEx§ (Physiological signal Explainer) is a novel Python library designed to address the black-box problem by providing a unified framework for both benchmarking and interpreting DL models for physiological signal analysis and sleep staging in particular. The toolbox enables users to (i) easily deploy DL models in a standard and deterministic environment with fixed seeds and dataset splits, and (ii) apply explainable AI (XAI) techniques specifically tailored to the sleep staging context to allow users, both developers and clinicians, to interpret the model behavior.

By offering a standardized environment for training and evaluation, PhysioEx allows users to make consistent and fair comparisons of the performance of custom models across various datasets. It also provides pre-trained versions of state-of-the-art models and helps developers shift the focus from merely achieving high performance to understanding how and why the performance is obtained.

Sleep staging has been identified as the main use case for PhysioEx due to the large amount of data available, the availability of well-performing DL models, and

§ `https://github.com/guidogagl/physioex` the well-documented ground truth (ie sleep stages) obtained according to the AASM standards[10].

Compared to other clinical tasks, the use of DL models[11, 12] in autonomous sleep staging is relatively new [2]. Initially, experts' knowledge was used to extract handcrafted features from EEG data and classify the sleep stages using machine learning approaches[13, 14]. DL models have been gradually introduced thanks to their ability to autonomously identify and extract these features[15], by employing neural networks predicting sleep stages for individual epochs[16, 17], i.e, 30 seconds of sleep data.

These DL models faced difficulties in capturing long-range dependencies between sleep epochs due to limited temporal context, crucial for accurate sleep staging [2]. Modern architectures with long-term modeling now model entire sleep cycles, enhancing robustness [18, 19].

Modern deep learning models for sleep staging are now part of the sequence-to-sequence framework[20], representing a significant step forward in the field. In this framework, the input is a sequence of consecutive epochs, and the output is a sequence of sleep stage classifications.

Because of this context, the application of XAI techniques[21] to sleep staging is particularly interesting and challenging when compared to other physiological signal analysis tasks. Moreover, the clearly defined rules in the AASM standard offer a strong validation potential for the explanations extracted from the models; while the same can't be applied to many other tasks, such as emotion-recognition[22] in which subjects perform self-labelling of the recorded data, and explainable DL is used as a knowledge discovery tool[23], i.e., to investigate the brain physiological correlations with the emotional labels inspecting the way the model is capable to match its inputs and outputs.

Popular explainable AI approaches on time-series analysis[24], such as attribution methods, focus on extracting the sub-part of the input signal which has the biggest influence on the prediction outcome. While this might be relevant to identify parts of a time series where particular waveforms like epileptic seizures [24]; sleep staging is less about detecting particular waveforms in the signal, but is often related to the time-invariant aspects of the signal, such as its spectral components[10]. For instance, sleep stages are typically identified by medical practitioners who leverage the presence of long wave patterns corresponding to the activation of specific frequency bands, such as Delta waves in NREM3 (Deep Sleep) stages. In this case, attributions methods may not fit the purpose of explaining the model decision-making system, and hence we aim to introduce new explainability approaches for such use cases.

Furthermore, state-of-the-art attribution methods do not directly support multi-output models such as modern sleep staging architectures (i.e., sequence-to-sequence). Transforming these models into their single-output versions (i.e. by focusing on one output at a time) overlooks the influence of attribution values on unobserved outputs.

Lastly, a general concern for XAI algorithms is that they are developed from by software developer and machine learning engineers, but results are not always aligned with the interpretation of end users and domain experts, e.g. physician and medical personnel.

For this reason, newer classes of XAI algorithms are under investigation, such as concept-based explanations methods[25, 26], which attempt to bridge this gap by extracting domain-related concepts from the model decision-making system to explain its outcome.

Concepts are designed to provide explanations aligned with the human expertise and are used by concepts-based-architectures[27, 28] to constraint the model to take decisions based on them. In sleep staging, the human expertise is outlined by the AASM manual, which defines the approach scorers follow while labelling the data. For example, the presence of K-complexes or sleep spindles typically indicates an NREM2 epoch, while delta waves are characteristic of Deep Sleep stages.

While promising, their implementation relies on the availability of concepts-labelled data, which are absent in most public datasets for sleep staging. For this reason, a challenge is to learn autonomously the concepts[29, 30] from big datasets or combinations of big datasets, validate the human acceptability of the learned concepts and finally use the learned concepts to label thousands or millions of sleep data, to enable the applicability of such concepts based architectures.

PhysioEx tackles these challenges by providing implementations of a range of post-hoc explainability methods. As such, it offers an easy way to explore the impact of time-invariant features of the input time-series on a chosen DL architecture, and new concept learning approaches capable of identifying concepts as elements of the network latent-space. Both these approaches proved their capabilities into sleep-staging tasks and aligned their results with the human expertise.

By doing this, PhysioEx helps sleep and DL researchers to move the focus from performance optimization to understanding the underlying mechanisms that contribute to model decisions. This shift in focus is crucial for fostering trust in AI models, particularly in clinical applications, where understanding the "why" and "how" behind model predictions is just as important as the accuracy of those predictions.

2. Materials and Methods

Sleep stages classification is a complex task that requires handling multichannel time-series data, preprocessing techniques, and the application of advanced deep learning models. PhysioEx addresses these challenges by providing an integrated toolkit that standardizes the entire workflow across different or custom architectures and datasets, from data preprocessing to model evaluation, and also offers explainability tools.

The list of the available datasets and models is available in the PhysioEx documentation|| and is continuously updated. Users can contribute to the library following the guidelines specified in the specific documentation page¶.

|| https://guidogagl.github.io/physioex/

¶ https://guidogagl.github.io/physioex/pages/contribute

The package is designed to be modular, allowing researchers to easily customize and extend functionalities according to their specific needs. In this section, we detail the design and implementation of PhysioEx, highlighting its key modules: *preprocess*, *data*, *train*, *explain*.

The library includes a set of pretrained models ready to be used on new data or as benchmark for newly introduced models. This will help users to focus on explainability research rather than performances.

Under the hood, PhysioEx is developed in Pytorch 2 and Pytorch Lightning, granting high compatibility with many other python oriented deep learning libraries such as Captum[31] for the explainability module.

At the moment, PhysioEx supports sleep staging using 3 state-of-the-art deep learning approaches and 6 sleep staging datasets.

2.1. The Preprocess Module

The module `PhysioEx.preprocess` is responsible for transforming raw sleep datasets into a format readable by PhysioEx, known as the PhysioEx dataset. PhysioEx offers extensive compatibility with datasets provided by the National Sleep Research Resource (NSRR)[7], as well as support for other publicly-available sleep staging datasets.

The supported datasets include:

- **SHHS (Sleep Heart Health Study)** [4] is a multicenter cohort study investigating the cardiovascular and other consequences of sleep-disordered breathing. At visit 1, it included 5,793 participants aged 40 years or older. PSG recordings were conducted in participants' homes by trained technicians, featuring C3/A2 and C4/A1 EEGs sampled at 125 Hz, right and left EOGs sampled at 50 Hz, and a bipolar submental EMG sampled at 125 Hz.
- **MASS (Montreal Archive of Sleep Studies)** [32] is an open-access collaborative database containing laboratory-based PSG recordings. It includes 200 complete nights recorded from 97 men and 103 women, aged 18 to 76 years. All recordings feature a sampling frequency of 256 Hz and an EEG montage of 4–20 channels, along with standard EOG and EMG.
- **MESA (Multi-Ethnic Study of Atherosclerosis)** [6] is a multicenter prospective study involving 2,237 ethnically diverse men and women aged 45 to 84 from six U.S. communities. PSG recordings were obtained in-home, including central C4-M1 EEG, bilateral EOG, and chin EMG sampled at 256 Hz. Scoring was performed by one of three certified polysomnologists.
- **MrOS (The Osteoporotic Fractures in Men Study)** [5] is a multicenter study comprising 2,911 PSG recordings from men aged 65 years or older, enrolled at six clinical centers. Recordings were conducted in home settings and included C3/A2 and C4/A1 EEGs, chin EMG, and left-right EOG, all sampled at 256 Hz.
- **HMC (Haaglanden Medisch Centrum)** [33] is a collection of 151 whole-night PSG recordings from 85 men and 66 women, gathered at the Haaglanden Medisch

Centrum sleep center. The data includes 4 EEG channels (F4/M1, C4/M1, O2/M1, and C3/M2), two EOG channels (E1/M2 and E2/M2), and one bipolar chin EMG, all sampled at 256 Hz.

- **DCSM (Danish Center for Sleep Medicine)** [34] consists of 255 randomly selected and fully anonymized overnight lab-based PSG recordings from patients seeking diagnosis for non-specific sleep-related disorders at the DCSM. The setup included EEG, EOG, and EMG channels, all sampled at 256 Hz.

PSG raw data is typically stored into `.edf` files for each recording, with or without the clinical annotations about the sleep stages. These files store the information in a standardized and easily accessible format, but are typically not suited to be used in DL scenarios, since the raw data needs further processing, segmentation and alignment with sleep labels to be provided as input to a DL model. To this end, PhysioEx creates a standard processing pipeline for all the datasets. The pipeline is detailed in Fig. 1.

While `.edf` files are a standardized data format for EEG and PSG data, their actual internal organization, e.g., channels naming, sampling frequency, annotations etc., largely differs considering different datasets. For this reason, PhysioEx lets users customize all the dataset-dependent information by extending few methods of the `PhysioEx.preprocess.Preprocessor` class, leaving the main logic to the module.

For each of these datasets, signals such as EEG, EOG, EMG, and ECG are extracted. The EEG signal is standardized to a single channel by selecting electrodes C3-M2 or C4-M1. If this setup is unavailable for a specific dataset or subject, PhysioEx automatically selects a backup setup. By extending the main module, the user can easily select different or new channels from the `.edf` files, depending on their availability in the desired dataset.

A tutorial on how to extend the `Preprocessor` class to integrate custom settings on already supported datasets or to integrate new datasets is available in the PhysioEx documentation$^+$.

The module sets up cross-validation by determining the number and composition of folds. For MASS and SHHS datasets, folds follow setups from [20] and [35]. Other datasets use PhysioEx's default strategy, randomly splitting subjects into train, validation, and test sets with 75% 15% 15% ratios, using a fixed seed to ensure consistency. Users can opt for the default or customized splits when adding new datasets.

The detailed default configurations for each dataset are provided in Table 1.

Once the library is installed, the module provides a command line tool:

```
preprocess --dataset [] --data_folder []
```

This command converts datasets to a PhysioEx-compatible format, enabling in-memory virtualization via indexing and memory-mapped files per subject. It includes an indexing table for data access, sub-directories for preprocessed sleep epochs as memmap

[+] `https://guidogagl.github.io/physioex/preprocess`

Dataset	EEG Channel	EOG	EMG	ECG	# Subjects
SHHS	C3-M2	✓	✓		5,793
MASS	C4-A1/C3-A2	✓	✓		200
MESA	C4-A1/C4-M1	✓	✓		2,055
MROS	C4-A1/C4-M1	✓	✓		2,898
HMC	C3-M2	✓	✓	✓	151
DCSM	C4-M1/C3-M2	✓	✓	✓	255

Table 1. EEG Channel Configurations for Supported Datasets

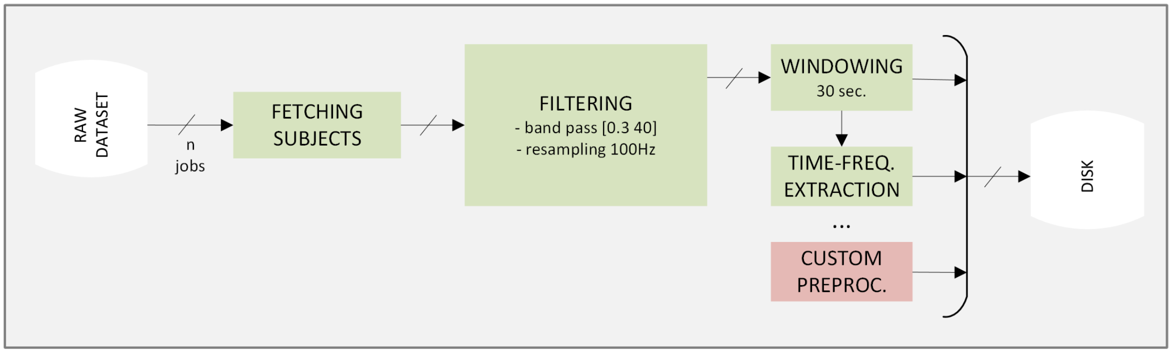

Figure 1. The preprocessing pipeline provided by the `PhysioEx.preprocess` module. Green boxes represent the default functionalities included in the pipeline: fetching subjects from the raw dataset, band-pass filtering and resampling at 100 Hz, windowing into 30-second segments, and time-frequency features extraction. Custom preprocessing steps, such as user-defined transformations, can be integrated where necessary, shown in the red box. The parameter `n_jobs` defines the number of parallel processes used to accelerate the pipeline execution, with the default value set to -1 for maximum parallelization. The processed data is stored on disk at the end of the pipeline.

arrays, and a labels directory. The indexing table also offers optional subject details like sex and age, which can be customized by the users if needed.

For publicly available datasets, such as DCSM and HMC, the library also handles the download of the dataset. The datasets available in the NSSR (SHHS, MESA and MROS) archive, instead, need to be downloaded with their command-line-interface and then placed into the specified data folder.

The module performs the preprocessing of the data in raw format and aligns it with the corresponding sleep stages. The outcome is the segmentation of each subject data into 30-second epochs, each associated with its sleep stage. The data is resampled to 100 Hz and bandpass filtered between 0.3 Hz and 40 Hz for EEG and EOG signals, and at 10 Hz for EMG signals. If wake epochs are overrepresented compared to sleep epochs, the module automatically reduces the number of wake epochs, cutting the initial and final part of the recording.

The module allows users to define custom preprocessors, which are callable methods applied to the sleep epochs to create additional preprocessed versions of the dataset. By default, PhysioEx computes the preprocessing proposed by [35], in addition to retaining the raw data. In this preprocessing, the 30-second time series are transformed into time-frequency images by dividing each epoch into two-second windows with 50% overlap, multiplied with a Hamming window, and transformed to the frequency domain using a 256-point Fast Fourier Transform; finally the amplitude spectrum is log-transformed.

2.2. Data Module

The module (`PhysioEx.data`) enables the serialization of preprocessed data from the `PhysioEx.preprocess` module into formats compatible with PyTorch 2 and PyTorch Lightning, enabling the training of PyTorch models.

PhysioEx supports sequence-to-sequence framework [2], requiring the data to be read and converted into sequences at runtime. Users can specify the desired length of the sequence to be provided to the model, and PhysioEx manages the sequential reading of sleep epochs from memory. Sleep epochs are finally standardized to 0 mean and unit standard deviation. Each of the subject recordings containing $num\_windows$ sleep epochs provides $num\_windows - sequence\_length + 1$ sequences of sleep epochs. Once fetched, each batch of raw data consists of a float tensor of shape $batch\_size \times sequence\_length$.

The data module is responsible for loading the data during training and evaluation supporting 2 circumstances: (i) the node RAM is lower than the dataset size (e.g., low-resources computing node); (ii) the system is running over a cluster of nodes in a distributed high-performance-computing (HPC) environment.

PhysioEx has been tested using servers with 8 cores, 16Gb of RAM and Tesla T4 GPU as low-resource nodes, and HPC infrastructures such as LUMI[*] and VSC♯.

2.2.1. Data loading on low-resources computing nodes DL models typically require loading the entire dataset into memory before dividing it into batches and swapping them into the GPU memory. The combined memory footprint of all supported datasets in PhysioEx is approximately 1.2 TB (and it is supposed to grow as new datasets will be added), implying that even high-resource compute nodes may struggle to handle all datasets in memory, particularly in cross-dataset training scenarios. Furthermore, training on large datasets like SHHS, which occupies more than 400Gb, can also be problematic over distributed HPC scenarios.

A straightforward solution involves training on batches of subjects loaded into memory, but this can cause the training algorithm to settle on local minima. The optimal approach should give the algorithm randomized data for each batch. PhysioEx solves this by using disk swapping to load only necessary sleep epoch sequences for each batch, allowing training even on low-resource systems.

[*] `https://www.lumi-supercomputer.eu`

[♯] `https://www.vscentrum.be`

This solution can be achieved using hierarchical data formats like HDF5, which allows the construction of a single file per dataset and the reading of file segments during batch loading. However, PhysioEx saves each subject's data as individual memory-mapped files, allowing partial file reading at runtime. To this end, PhysioEx builds an index mapping each element of the dataset to its subject-file, keeping the read-complexity to $O(1)$.

This method is favored over hierarchical formats for handling large datasets on shared file systems like GlusterFS or NFS, as a single file can bottleneck data transfers, reducing performance and training speed. In PhysioEx, RAM usage is more influenced by worker count per PyTorch `DataLoader` than batch size, with each worker using a prefetching queue for batch building and GPU transfer. GPU RAM depends on batch size, while CPU RAM usage is tied to how quickly the prefetching queue fills, influenced by worker number. Matching workers to CPU cores is ideal, but may increase RAM usage on multi-core systems. PhysioEx allows customization of workers per dataloader.

Increasing the number of workers speeds up training and evaluation because of better parallelization of data loading. However, since this is an I/O-bound process, the speed gain depends on the disk's read speed, and too many workers may only add unnecessary overhead without further performance improvement.

2.2.2. Data loading on high-performance-computing clusters PhysioEx supports HPC and distributed systems training with distributed-data-parallel for multi-GPU or multi-node setups. Users can configure node numbers and use a flag for slurm or torque clusters. In distributed training, datasets are sharded across nodes, each handling a batch, with gradients aggregated cluster-wide. Data loading can be challenging without ample RAM; for example, loading 1.2TB requires at least 10 nodes with 128GB RAM each, increasing scheduling time. PhysioEx uses disk-memory swapping, indexing data on nodes and lazily loading it to avoid overwhelming memory.

2.2.3. Data loading on combined datasets The data module also allows an easy merging of multiple datasets into a unified dataset. This functionality is supported by easily passing a list of the dataset names to be combined to the data loading module. Thanks to this, PhysioEx supports cross-datasets training scenarios.

To achieve this, PhysioEx constructs a level-1 index, mapping each element of the combined dataset to its specific index within the individual dataset.

Each fetch cycle for a sequence of epochs of dimension L with absolute index in the combined dataset $indx$ includes: accessing the level-1 index at position $indx$ to retrieve the dataset (complexity $O(1)$); accessing the level-2 index at position $indx$ to retrieve the subject identifier (complexity $O(1)$); accessing the level-3 at position $indx$ to retrieve the position of the first epoch of the sequence in the memory mapped file (complexity $O(1)$); read from the disk L elements from the data file and from the labels file (complexity $O(L) + O(L)$).

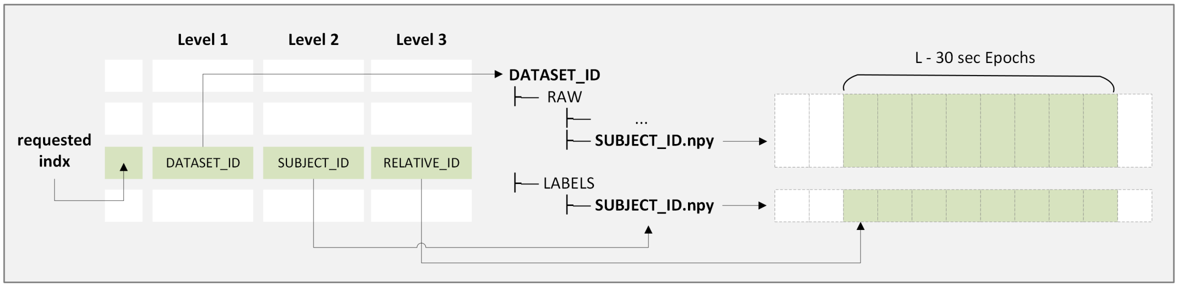

Figure 2. Illustration of the multi-level indexing approach used in the data module of PhysioEx for retrieving sequences of L epochs. The level-1 index maps the unified dataset to individual datasets, providing an index to each dataset. The level-2 index maps each dataset entry to a specific `subject_id`. Finally, the level-3 index identifies the starting position of the first epoch within a sequence of L epochs in the data of the given `subject_id`. This hierarchical indexing structure enables efficient $O(1)$ retrieval of epoch sequences across multiple subjects and datasets.

The overall fetch cycle computational complexity is $O(1) + O(1) + O(1) + O(L) + O(L) = O(L)$ if L is small compared to the number of instances of the dataset, we can say that $O(L) \approx O(1)$. The fetching pipeline used by PhysioEx is detailed in Fig. 2.

This complexity is not affected by the number of datasets included. Hence, as the number of instances in the merged datasets is equal to the sum of the instances of each dataset n, reading all the merged dataset instances has complexity $n * O(1) = O(n)$, which is optimal.

As the RAM occupation does not depend on the number of instances n, but only on the number of workers provided by the user, PhysioEx allows loading datasets or combinations of datasets with minimal resources, increasing the fetching time due to the disk-swapping overhead.

An option to improve data-fetching time is to use popular memory caching systems, such as Least Recently Used (LRU) caching. However, since the entire dataset is typically accessed sequentially during training, with different sequences being fetched each time, this caching system offers no benefits, and increases the RAM occupation.

Alternatively, caching all data for a subject when fetching a single sequence could be considered, but since the fetch cycle has a time complexity of $O(1)$, this method wouldn't provide any meaningful improvement.

2.3. The Train Module

The `PhysioEx.train` module in PhysioEx provides a standardized training interface based on PyTorch Lightning for training PyTorch models.

Before any training or testing operation, PhysioEx ensures reproducibility by setting a fixed seed for all pseudo-random number generators used in PyTorch, NumPy, and Python's built-in random module. Additionally, all PyTorch DataLoader workers are seeded to guarantee consistent data shuffling and batching across different runs.

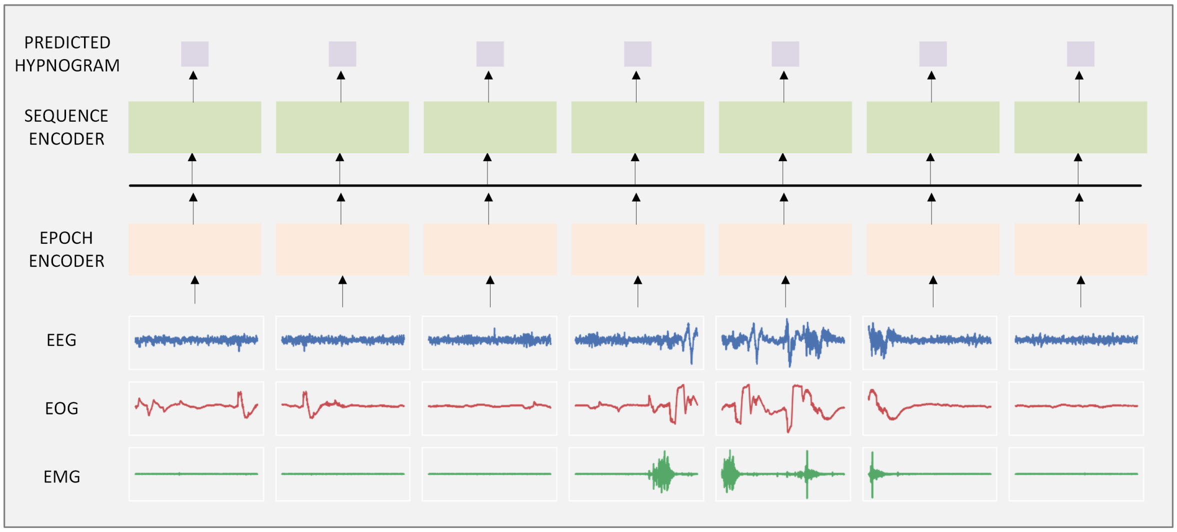

Figure 3. Schematic representation of a sequence-to-sequence model architecture as implemented in PhysioEx. The model consists of two main components: an epoch encoder and a sequence encoder. Each input epoch within a sequence of L epochs is processed by the epoch encoder to produce L epoch encodings. Sequences can be either single-channel or multi-channel (i.e. in the feature the encoder takes EEG, EOG and EMG data). These encodings are then passed through the sequence encoder, which generates L sequence encodings. Finally, each sequence encoding is classified into a corresponding sleep stage. PhysioEx allows users to easily create custom sequence-to-sequence models by specifying the implementation of the epoch encoder and sequence encoder.

PhysioEx supports models compatible with the sequence-to-sequence framework. In this framework, a model M comprises an epoch encoder and a sequence encoder. Given an input sequence of L epochs, each epoch is encoded by the epoch encoder to produce L epoch encodings. These encodings are then processed by the sequence encoder, which produces L sequence encodings that are subsequently classified into L sleep stages by the model.

A sequence-to-sequence model is generally a Multi-Input Multi-Output (MIMO) model that takes L epochs as input and produces L classifications as output, one per epoch. In addition, PhysioEx supports sequence-to-epoch models, or Multi-Input Single-Output (MISO) models, where the model takes a sequence of L epochs as input and predicts the sleep stage of a specific epoch within the sequence, such as the central epoch in the Chambon2018 model[36].

PhysioEx provides PyTorch 2 implementations of three state-of-the-art architectures for sleep stage classification: TinySleepNet[37], SeqSleepNet[20], and Chambon2018[36]. All of these approaches have been implemented and tested, leading to results in-line with the results in the original publications, detailed in Tab 2.

Chambon2018 is a sequence-to-epoch model which consists of one epoch-encoder implemented by a convolutional neural network (CNN) model that processes raw sleep data followed by ReLU activations and max-pooling layers. These operations capture local temporal and frequency patterns in the input signals. The features extracted by the epoch-encoder are then flattened and passed through a fully connected layer for the classification of the central epoch within the sequence. Chambon2018 is the smallest network available in PhysioEx with 17.9K parameters.

SeqSleepNet is a sequence-to-sequence two-stage recurrent neural network (RNN) designed to capture both short-term and long-term temporal dependencies in the input data. It takes time frequency images as input. The epoch encoder uses a bi-directional LSTM (32 hidden units per direction) to model the dependencies within each 30-second epoch, followed by an attention mechanism that sums the temporal learned contributions. The sequence encoder employs a GRU (with 128 hidden units) to process the sequence of encoded epochs over time, capturing long-term temporal relationships across multiple sleep epochs. The output is passed through a fully connected layer to classify the sleep stage. This dual-layered RNN architecture allows SeqSleepNet to efficiently model both short-term (within an epoch) and long-term (across epochs) temporal structures.

TinySleepNet is a sequence-to-sequence model taking raw time-series as input. The model includes an epoch-encoder based on a stack of 1D convolutional layers with batch normalization and ReLU activation. The convolutional layers extract spatial-temporal features from the input by applying different kernel sizes. A final flatten layer reduces the dimensionality of the extracted features before passing them through a bi-directional LSTM (128 hidden units per direction) for sequence processing. A linear classifier is applied at the output to predict the sleep stage. TinySleepNet consists of 1.5M parameters which is significantly higher than SeqSleepNet which consists of 137K parameters. On the other side, TinySleepNet processes raw signals as input while SeqSleepNet needs first to transform signals in the time frequency domain before providing them as input.

To be mentioned, a fair comparison between these models relying only on the published performances could be troublesome since they are often evaluated on different datasets, using varying seeds, splits, and parameter configurations. PhysioEx provides a unified framework for consistent benchmarking, allowing researchers to test these models under standardized conditions.

These three models were chosen for the first release of PhysioEx because they represent different approaches to sleep staging. TinySleepNet processes raw signals, while SeqSleepNet uses time-frequency images, both using a sequence-to-sequence framework. Chambon2018, on the other hand, does not rely on this framework but is still important because most Explainable AI methods are designed for single-output models, which makes it easier to apply explainability techniques to Chambon2018 compared to multi-output models.

The Train module implements a command-line-interface (CLI) allowing users to quickly set up the train and evaluation of a sleep staging model on different datasets.

```
train --model [] --datasets []
```

Network	**Year**	**Input**	**Epoch Encoder**	**Sequence Encoder**	**N°Params.**	**MASS Acc.**
Chambon2018	2018	Raw	CNN	Dense	17.9K	79.9%
SeqSleepNet	2018	Time-Freq.	RNN	RNN	137K	87%
TinySleepNet	2020	Raw	CNN	RNN	1.5M	82.6%-87.5%

Table 2. Summary of key characteristics of the three neural networks used for sleep stage classification available in PhysioEx. The table presents the year of publication, input type (raw signals or time-frequency representations), architecture of the epoch encoder and sequence encoder, number of trainable parameters, and the mean overall accuracy achieved on the MASS[32] dataset as released by the authors in [36, 20, 37]. TinySleepNet mean accuracy refers to the worst and best performing partition of the MASS dataset: SS01 82.6% and SS03 87.5%.

The *train* command enables training a PyTorch 2 model on the specified list of datasets (merged together in case more than one dataset is provided). Users can provide their custom model implementations in a `.yaml` file for training it. Models in PhysioEx are evaluated using a set of standard metrics for sleep staging, including accuracy, Cohen's kappa, precision, recall, and F1-score[35]. During training, PhysioEx saves the best performing model as a checkpoint, defined as the one that maximizes the evaluation metrics on the validation set (i.e., it minimizes the validation loss).

The Train module also provides an interface for fine-tuning and testing a pretrained model, mirroring the training command, which allows users to adapt or evaluate existing models on new data. Custom models can be provided both using a `.yaml` configuration file, if the model class is not explicitly supported by PhysioEx, or by specifying a checkpoint file to load the model from.

```
test_model --model [] --datasets []
finetune --model [] --datasets [] --learning_rate []
```

The `PhysioEx.train` module offers an API for loading pretrained models. PhysioEx includes two pretrained versions of each implemented network, trained on the SHHS dataset: one version trained with a single-channel EEG configuration and another with multi-modal EEG, EOG, and EMG channels. Each of the pretrained versions available on PhysioEx has been trained to classify sequences of 21 sleep epochs, as described in [20, 35, 37, 36].

2.4. The Explain Module

PhysioEx makes sleep staging models accessible and readily available, facilitating their benchmarking. This allows to shift the focus to the last important aspect of the toolbox: model explanations.

In Python, most state of the art explainable AI algorithms are available through the Captum library[31], which PhysioEx directly supports, being built upon Pytorch 2. Some of the most popular interpretation approaches Captum provides are Integrated

Gradients (IG)[38] and Expected Gradients (EG)[39]. Both of these algorithms are gradient-based attribution methods, i.e., considering a model classifying an input instance x in a specific class, they attribute an importance value to each element x_i of x, corresponding to the i-time point in the input signal.

IG is a path-based method that assigns importance scores to input features by integrating the gradients of the model's output with respect to the inputs, along a path from a baseline (typically zero, as suggested by the authors in [38]) to the input itself. EG, on the other hand, extends IG by incorporating a probabilistic view. Instead of selecting a single baseline, EG averages over a distribution of baselines, thereby reducing the potential noise introduced by an arbitrary choice of the baseline.

As already discussed, these attribution methods may be not suited for sleep staging tasks, in particular when the model should rely on time-invariant features of the input and not on specific patterns.

For this reason, PhysioEx introduces Spectral Gradients, a novel attribution method which spreads the attributions scores on the frequency and time domain at the same time. Doing this, Spectral Gradients is able to identify relevant spectral components of a time-series when classifying instances not based on specific patterns but on time-invariant features. Considering an input signal x consisting in 3000 time-points, using Spectral Gradients, the user can divide the input signal spectrum, i.e., $[0, fs/2)$, into n partitions, and the algorithm will return a matrix of $n \times 3000$ attributions, each element of the matrix i, j represent the importance value of the i-band at time j. The algorithm also allows retrieving the importance attributions for each band, ignoring the time-resolution. Interested readers can interact with Spectral Gradients explanations using the jupyter notebook provided in the examples of the library.

A final contribution of PhysioEx is the implementation of a novel semi-supervised concept-learning method, allowing users to enrich datasets with conceptual information.

In PhysioEx, concepts serve as foundation cases for the sleep staging task, therefore they are input elements (i.e., time-series or time-frequency images) that are highly typical of one sleep stage and the underlying data distribution. These concepts should be input elements that human experts can easily comprehend. The actual input data should share some similarities with concepts; the closer the actual input is to the concept, the easier it is to assign it to a sleep stage.

The autonomous concept-learning approach, implemented into PhysioEx, consists of a sequence encoder-decoder architecture inspired by SeqSleepNet trained jointly in a self-supervised task (i.e. input reconstruction) and a prototype learning[30] task. Once trained, the learned prototypes are reconstructed into the input space and treated as concepts by the system. By doing this, the concept-learning algorithm is capable to assign conceptual information to each input element of the network by evaluating their proximity to the learned prototypes in the network latent space. In this way, PhysioEx is capable of enriching the dataset information with conceptual information. An overview of this system is presented in Fig. 4.

The technical details of the explanation approaches employed by PhysioEx are

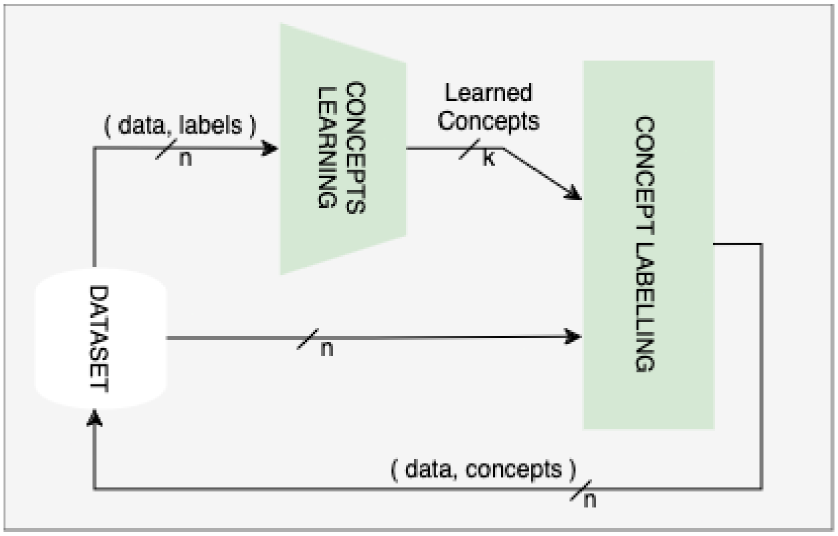

Figure 4. Overview of the concept labelling tool available into PhysioEx. The system is composed by a deep learning semi-supervised model that, given a dataset, learns a set of k-concepts. Each of these concepts represents a mode of the sleep staging task, i.e., a base case of the target problem solution. These learned concepts should be then physiologically validated, assessing their clinical relevance, by human experts. Once this is done, the library offers a concept-labelling tool which provides k-membership scores to each data available into the dataset, and store this information.

currently confidential and will be further documented in forthcoming publications and will be fully released in future micro-releases of the library upon the completion of the publication process.

3. Results

This section details the experimental results obtained using the models and the methodologies available in PhysioEx. The section is divided into two parts: in the first, we present a comprehensive benchmark of the pretrained models available in PhysioEx across multiple datasets. This benchmark aims to evaluate the generalizability and performance of state-of-the-art deep learning models for physiological signal analysis when tested on different datasets, using a standardized framework for dataset splits, parameterization, and evaluation metrics; the second part focuses on the explanation methods integrated into PhysioEx presenting the results of applying several explainable AI (XAI) techniques to the physiological signal models, demonstrating their ability to provide interpretable and clinically relevant insights into their decision-making processes.

3.1. Benchmarking results

During the benchmarking experiments, we evaluated the performance of the models described in section 2.3. These models were tested across the datasets in two

	Model	Input	#Chan.	Dataset	Acc	F1	CK	Prec.	Rec.
Train	SeqSleepNet	Time-Freq.	1	shhs	87.3%	86.9%	82.1%	86.9%	87.3%
			3		88.3%	88.0%	83.6%	87.9%	88.3%
	Chambon2018	Raw	1		84.5%	83.8%	78.3%	84.1%	84.5%
			3		87.1%	86.7%	81.8%	87.0%	87.1%
	TinySleepNet	Raw	1		87.7%	87.2%	82.6%	87.2%	87.7%
			3		89.1%	88.9%	84.7%	88.8%	89.1%
Test	SeqSleepNet	Time-Freq.	1	mros	70.5%	70.1%	56.3%	72.4%	70.5%
				mesa	59.0%	57.3%	42.2%	63.5%	59.0%
				dcsm	64.6%	66.4%	48.4%	77.5%	64.6%
				hmc	71.1%	68.9%	61.3%	72.0%	71.1%
				mass	78.0%	76.7%	68.2%	79.7%	78.0%
			3	mros	76.5%	76.9%	65.4%	79.7%	76.5%
				mesa	70.7%	70.1%	58.3%	73.6%	70.7%
				dcsm	60.2%	62.4%	43.3%	77.1%	60.2%
				hmc	65.7%	63.6%	54.2%	67.5%	65.7%
				mass	78.9%	77.8%	70.0%	80.8%	78.9%
	Chambon2018	Raw	1	mros	61.4%	59.9%	42.6%	64.0%	61.4%
				mesa	21.6%	19.4%	7.2%	36.9%	21.6%
				dcsm	52.6%	54.4%	29.3%	64.5%	52.6%
				hmc	69.7%	67.1%	60.2%	70.1%	69.7%
				mass	79.1%	77.0%	70.1%	79.4%	79.1%
			3	mros	61.2%	61.1%	45.4%	70.6%	61.2%
				mesa	22.4%	20.1%	8.5%	38.0%	22.4%
				dcsm	38.6%	42.6%	16.8%	61.3%	38.6%
				hmc	66.4%	65.1%	55.7%	66.5%	66.4%
				mass	70.0%	68.9%	59.8%	75.6%	70.0%
	TinySleepNet	Raw	1	mros	68.7%	66.5%	52.1%	67.9%	68.7%
				mesa	38.0%	22.3%	-2.1%	19.0%	38.0%
				dcsm	57.9%	44.8%	-0.0%	37.8%	57.9%
				hmc	67.3%	64.6%	56.2%	67.6%	67.3%
				mass	80.9%	79.3%	72.5%	80.2%	80.9%
			3	mros	71.8%	70.0%	57.2%	73.7%	71.8%
				mesa	30.5%	21.8%	-0.7%	19.3%	30.5%
				dcsm	33.4%	34.0%	8.7%	51.0%	33.4%
				hmc	66.1%	63.3%	54.5%	66.0%	66.1%
				mass	76.9%	75.9%	66.8%	77.1%	76.9%

Table 3. Performance results on the test splits of various datasets for pretrained models available in PhysioEx. The models were pretrained on the training split of the SHHS dataset. Each model was evaluated using different input channel configurations (1 or 3 channels), and standard sleep stage classification metrics are reported, including accuracy (Acc), F1 score (F1), Cohen's Kappa (CK), precision (Prec.), and recall (Rec.).

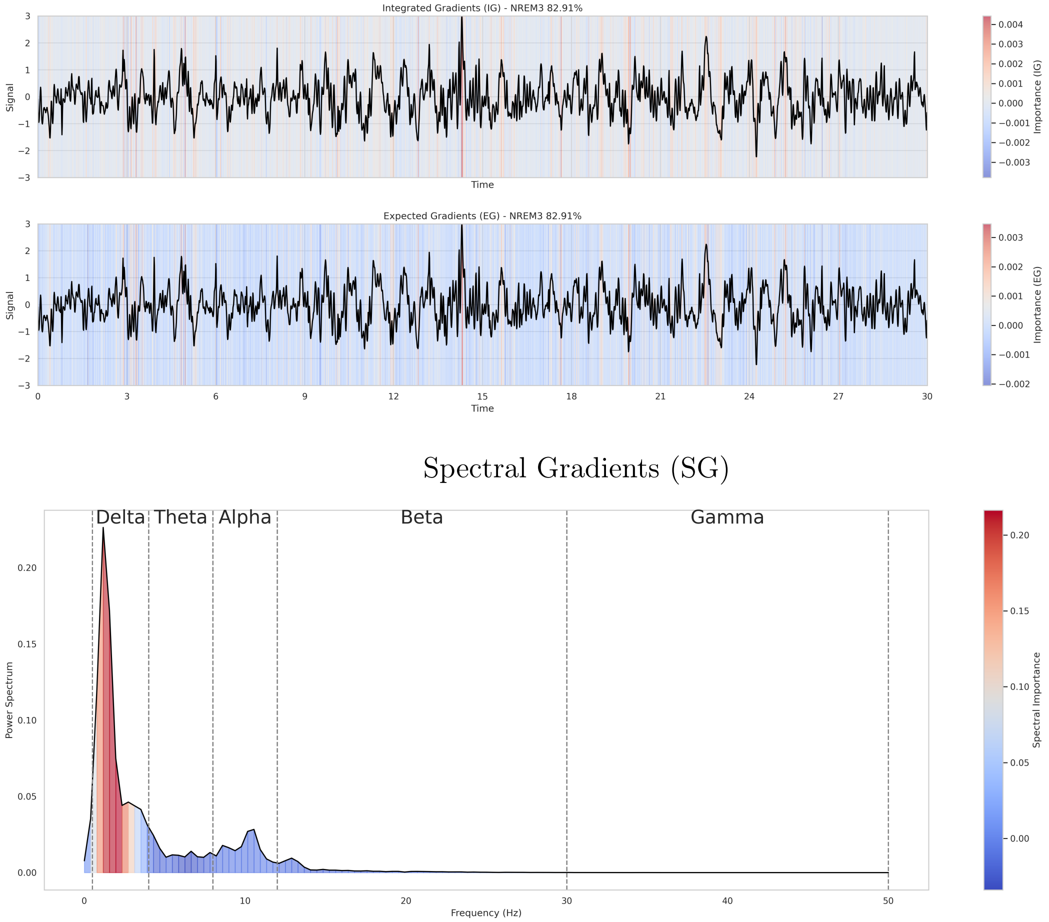

Figure 5. Illustration of an EEG epoch of sleep, correctly classified by the Chambon2018 model as NREM3, i.e, Deep Sleep. The picture reports the confidence of the predicted class.

In the first two pictures, the overlaid red-to-blue color scale represents the importance values extracted using two state-of-the-art attribution algorithms. The intensity of the color corresponds to its relative importance, with red indicating higher positive attribution and blue indicating negative attribution. In these pictures, there is no clear temporal pattern in the signal where a specific segment stands out as having higher importance (red overlay). This suggests that the model likely leverages time-invariant components of the signal, which are not captured by state-of-the-art XAI methods.

The 3^{rd} figure shows the power spectrum of the signal across different frequency bands (Delta, Theta, Alpha, Beta, Gamma), alongside the spectral importance attributed by the PhyioEx Spectral Gradients algorithm to each frequency. Notably, the Delta band, which is characteristic of deep sleep, is correctly identified as the most important in the NREM3 epoch. Note that the method has no awareness of the sleep bands, but autonomously detects the most relevant part of the spectrum for the classification outcome.

Model	Input	#Chan.	Dataset	Acc.	F1	CK	PR	RC
SeqSleepNet	Time-Freq.	1	mros	87.05%	86.45%	80.34%	86.33%	87.05%
			mesa	83.52%	82.96%	75.98%	83.03%	83.52%
			dcsm	89.51%	88.95%	81.69%	89.09%	89.51%
			hmc	77.66%	77.46%	70.41%	77.66%	77.66%
			mass	85.10%	84.70%	78.01%	84.88%	85.10%
		3	mros	88.04%	87.47%	81.86%	87.32%	88.04%
			mesa	85.88%	85.42%	79.53%	85.42%	85.88%
			dcsm	90.19%	89.64%	83.01%	89.79%	90.19%
			hmc	75.02%	74.83%	67.30%	75.12%	75.02%
			mass	86.73%	86.29%	80.30%	86.51%	86.73%
TinySleepNet	Raw	1	mros	87.25%	86.66%	80.63%	86.61%	87.25%
			mesa	83.84%	83.29%	76.49%	83.39%	83.84%
			dcsm	89.82%	89.53%	82.38%	89.69%	89.82%
			hmc	78.42%	78.13%	71.55%	78.14%	78.42%
			mass	86.42%	86.08%	80.05%	86.19%	86.42%
		3	mros	88.32%	87.68%	82.24%	87.61%	88.32%
			mesa	86.38%	86.20%	80.34%	86.33%	86.38%
			dcsm	90.61%	90.50%	83.92%	90.87%	90.61%
			hmc	77.17%	76.75%	69.80%	76.96%	77.17%
			mass	88.49%	88.25%	83.09%	88.31%	88.49%
Chambon2018	Raw	1	mros	84.05%	82.75%	75.61%	83.02%	84.05%
			mesa	79.97%	78.28%	70.34%	79.04%	79.97%
			dcsm	86.51%	84.86%	75.93%	84.58%	86.51%
			hmc	75.48%	74.55%	67.59%	75.07%	75.48%
			mass	84.44%	83.55%	77.00%	84.29%	84.44%
		3	mros	86.91%	85.90%	79.99%	86.13%	86.91%
			mesa	84.10%	83.11%	76.71%	83.51%	84.10%
			dcsm	88.08%	86.74%	78.94%	86.82%	88.08%
			hmc	74.62%	73.83%	66.31%	74.40%	74.62%
			mass	86.03%	85.32%	79.53%	86.27%	86.03%

Table 4. Performance results for the pretrained models shown in Tab 3 in a fine-tuning experiment. The models have been fine-tuned with all their parameters in training mode, learning rate equals to $1e-4$ and 5 epochs.

configurations: unimodal (single-channel EEG) and multimodal (using EEG, EOG, and EMG). The datasets used for evaluation include SHHS, MROS, MESA, DCSM HMC, and MASS, which are all supported by PhysioEx.

In the first experiment, all models were trained on the training split of the SHHS dataset and then tested on both the test split of SHHS and the test splits of the other datasets. Table 3 summarizes the performance metrics considering the mean accuracy (Acc), F1-score (F1), Cohen's Kappa (CK), precision (Prec.), and recall (Rec.), for each model and dataset.

Considering Table 3, TinySleepNet achieves the highest performance when trained

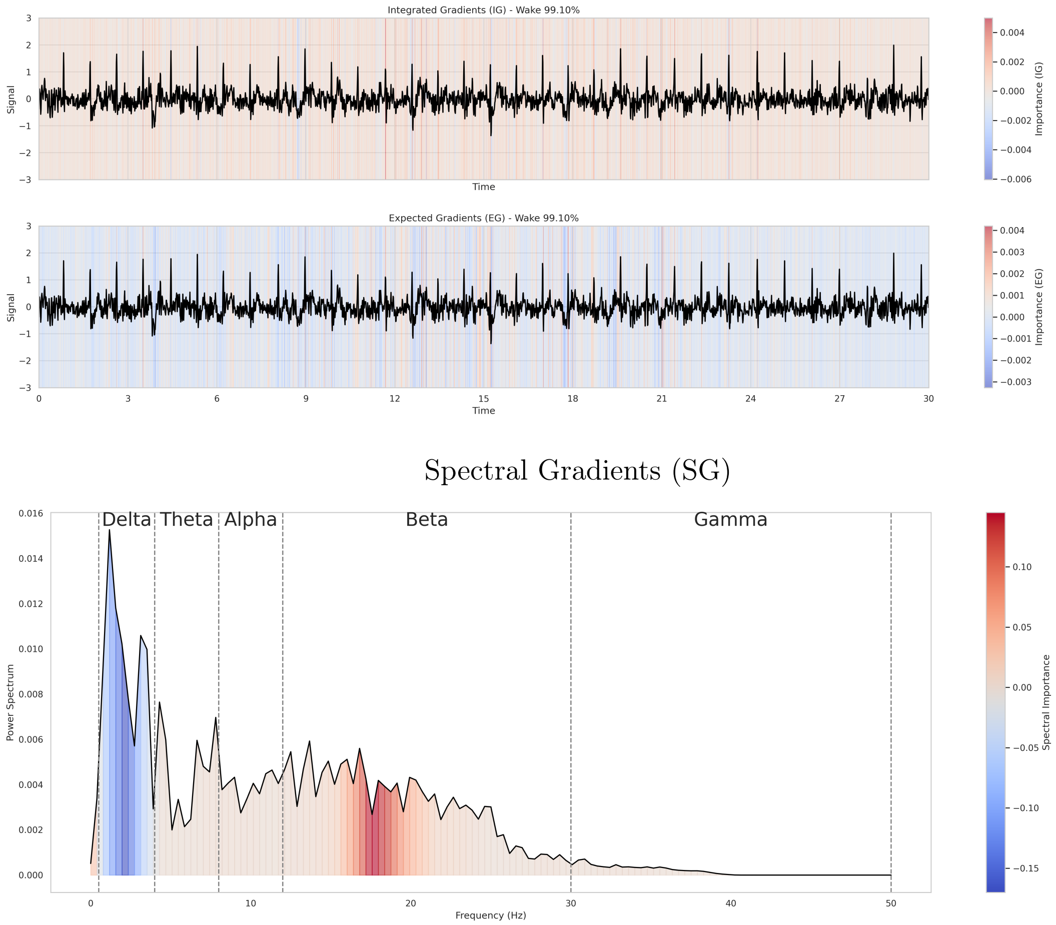

Figure 6. Outcome of the PhysioEx Spectral Gradients method compared to two-state-of-the art attribution methods applied to the same sleep epoch, correctly classified by the Chambon2018 model as Wake. The figure presents the same setup of Fig. 5, but for a Wake stage instead of a NREM3 stage.

In this case, Spectral Gradients (SG) correctly identifies the Beta bands as the more important, which are characteristic of Wake stages. Note that SG correctly assigns negative importance to Delta bands, even though Delta has maximum power in the signal spectrum, i.e., SG assigns the correct importance to a sleep band independently of the band power.

This demonstrates the capability of PhysioEx to more accurately highlight relevant spectral components, particularly in sleep staging tasks. In contrast to the results of state-of-the-art attribution methods, where no distinct pattern emerged.

and tested on the SHHS dataset, with an accuracy of 89.1% in the multimodal configuration (3 channels) and a Cohen's Kappa (CK) of 84.7%. SeqSleepNet and Chambon2018 models also have comparable perfomances, particularly in their multimodal configurations, with SeqSleepNet reaching 88.3% accuracy and 83.6% CK, and Chambon2018 reaching 87.1% accuracy and 81.8% CK. In the single-channel EEG configuration, the performance gap between the models is smaller, with TinySleepNet still leading with 87.7% accuracy and 82.6% CK, closely followed by SeqSleepNet with

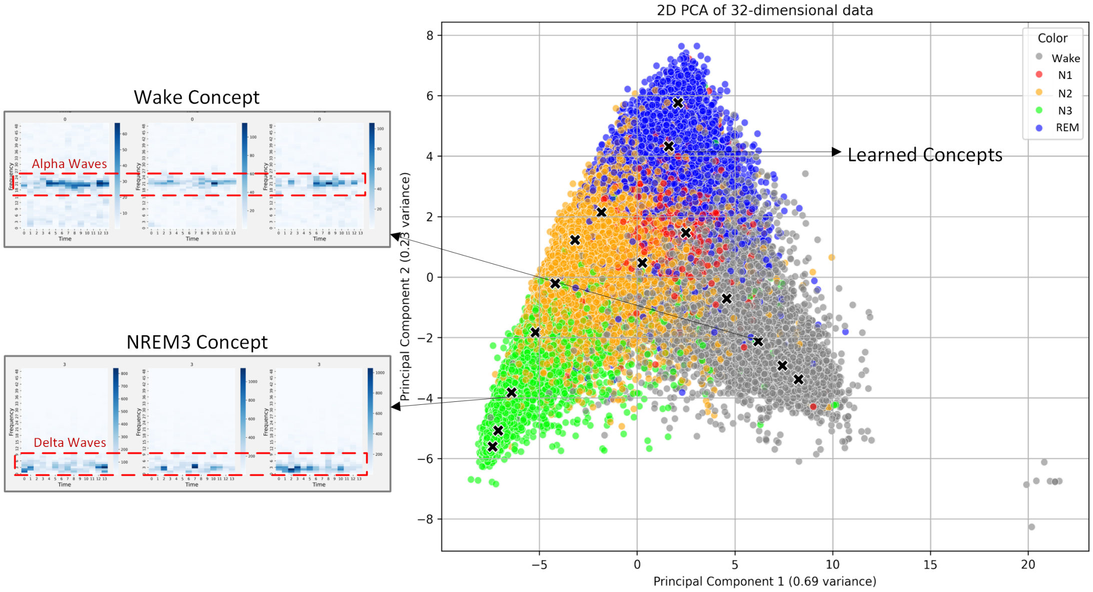

Figure 7. Illustration of a concept-learning approach available in PhysioEx. The approach learns time-frequency images with sequence length equal to 3 as concepts. On the right, the plot shows the first two principal components of the learned conceptual space. The space is composed by the latent-space-projections of sleep sequences of the SHHS dataset in the concept learning architecture.

The conceptual space highlights the differentiation of sleep stages (i.e., the color differentiation). In such space, the method identifies the concepts, i.e., points marked with x. These points are reconstructed from the latent-space to the input space, and their physiological relevance for sleep staging is analyzed.

For instance, the learned Wake stage concept highlites the presence of Alpha waves, and the learned NREM3 stage concept hihglites the presence of Delta waves. This demonstrates the system's capability to associate the correct physiological behavior with its corresponding sleep stages.

87.3% accuracy and 82.1% CK. Chambon2018 performs less well in this configuration, achieving 84.5% accuracy and 78.3% CK.

To be noticed, the performances obtained by SeqSleepNet resemble the one released by the authors on the same dataset in [35].

The pretrained models are then tested on the test splits of the other datasets (MROS, MESA, DCSM, HMC, and MASS), to obtain their generalization capabilities.

Considering the two biggest datasets (MROS and MESA), SeqSleepNet consistently outperforms TinySleepNet obtaining on average 73.6% vs 51.2% accuracy on the 3 channels setting and 64.8% vs 53.35% in the one channel setting; with Chambon2018 obtaining 41.8% and 41.5% in the 3 and 1 channels setting respectively. This trend is also observable aggregating the metrics of all the other smaller testing datasets, with SeqSleepNet sharing the best generalization performances with 70.4% accuracy on average on the 3 channels setting and 68.6% on the single channel setting, TinySleepNet reached 55.7% and 62.6%, while Chambon2018 obtained 51.7% and 56.9%.

Chambon2018, being a MISO (sequence-to-epoch) model is not fully comparable with the other setups, since it predicts the central epoch of the sequence only, but generally underperforms in comparison to the other sequence-to-sequence models.

To evaluate the applicability of these pretrained models into a real-world setting, we evaluated them in a fine-tuning experiment using the train-validation split of the forementioned test datasets.

In a real world scenario, we typically learn from a small amount of data provided by individual and even small clinics, with constrained amount of training time due to the costs associated to the rent of the GPU nodes.

Fine-tuning refines pre-trained models on smaller datasets by adjusting their weights while preserving general knowledge. This reduces overfitting and allows rapid deployment due to fewer training epochs, as foundational features are already learned and only specific adjustments are needed.

In our experiment, we retained all network parameters in training mode, with the learning rate at $1 - e4$, matching the training experiment. We fine-tuned the models for only 5 epochs, compared to 100 epochs needed in training, balancing performance and specialization time.

On the biggest datasets (MESA and MROS) the overall fine-tuning experiment (training and evaluation) on 5 epochs needed on average 8-16 hours respectively, setting the batch size equal to 256, on 2 nodes with 2 GPUs each in a torque cluster equipped with NVIDIA A100 GPUs.

The results of the fine-tuning experiment are detailed in Table 4. Considering the two biggest datasets, TinySleepNet slightly outperforms SeqSleepNet obtaining on average 87.4% vs 87% accuracy on the 3 channels setting and 85.6% vs 85.3% in the one channel setting; with Chambon2018 obtaining 85.5% and 82% in the 3 and 1 channels setting respectively. This trend is also observable aggregating the metrics of all the other smaller testing datasets, with TinySleepNet reaching 86.2% accuracy on average on the 3 channels setting and 85.2% on the single channel setting, SeqSleepNet obtained 85.2% and 84.6%, while Chambon2018 obtained 84% and 82.1%.

The fine-tuning experiment confirmed that sequence-to-sequence models outperform sequence-to-epoch models across different datasets. Both TinySleepNet and SeqSleepNet performed similarly, with TinySleepNet being slightly better on smaller datasets. However, TinySleepNet has significantly more parameters than SeqSleepNet (as shown in Table 2), making it more prone to overfitting, which can lead to better training results but poorer generalization. SeqSleepNet, with fewer parameters and time-frequency image inputs, demonstrated better generalization on larger datasets, supporting the results shown in Table 3.

3.2. XAI results

In this section, we present the experimental results on the interpretability of the sleep staging architectures available in PhysioEx.

First, we considered the pretrained single-channel Chambon2018 model. As previously discussed, while this model is less effective in predicting sleep stages compared to other state-of-the-art approaches, it offers the advantage of being a single-output model. This makes it more straightforward to interpret and explain using current state-of-the-art explainable AI algorithms, providing an ideal starting point for examining the decision-making process behind sleep stage classification.

To this aim, we tried to explain the model using the Captum implementation of Integrated Gradients and Expected Gradients (Section 2), to extract local explanations in a sleep-staging task, and then we compared it with an alternative approach proposed in PhysioEx, i.e., Spectral Gradients. Since Chambon2018 classifies the central epoch of the sequence provided as input, we considered sequences of epochs correctly predicted as NREM3 (i.e., deep-sleep) and we focused on obtaining local explanations of the central epoch.

We visualized the feature importance attributions obtained using both methods. The results, as shown in Fig 5 and Fig 6, detail how gradient-based attribution methods are not effective in capturing importance values when these are not related to specific patterns or time-variant features of the signal[24]. In fact, in the figures we can see how the importance values are very smoothed and spread over the time-series, not indicating a specific temporal component which is driving the classification outcome.

We then tested the same model and inputs using Spectral Gradients with n=50 linear partitions of the spectrum (i.e., each portion has 1Hz resolution). The aggregated results on the spectral components are shown in Fig 5 and Fig 6. Spectral Gradients correctly capture the most relevant information for the classification of the NREM3 and Wake sleep stage (the high power of Delta bands in NREM3 stages and the presence of Beta bands in Wake[10]) associating to that part of the spectrum the positive gradients i.e., the gradients that increase the confidence of the target class.

Further visualizations of the difference between IG, EG and SG are available in the examples' notebooks†† of PhysioEx.

For the second experiment, we tested a novel concept-learning approach implemented into PhysioEx based on SeqSleepNet, with a sequence length $L = 3$ on the SHHS dataset.

The results, shown in Figure 7, highlight the capability of the model to cluster similar sleep stages into well-defined regions within the latent space and to learn, as concepts, elements inside these well-defined regions (marked with x in the plot). The learned concepts can be decoded from the latent space to the input space. In figure 7 two concepts of NREM3 and Wake stages are decoded into time-frequency images. The decoded images correctly highlight the presence of Delta waves in NREM3 stages and

†† `https://github.com/guidogagl/physioex/blob/main/examples/spectral\_gradients.ipynb`

Alpha waves in Wake stages.

Once these concepts are validated, the system can assign to the input data a score equals to its membership function on a specific concept. This function is evaluated considering the distance between the latent-space-projection of the input and the concept.

The main limitation of the approach lies in its convergence for low-context scenarios only, i.e., when a limited number of epochs compose a sequence. However, this limitation will be addressed in the next micro-release of the library, which will include improvements to enhance performance in higher-context situations.

4. Discussion and Conclusion

In this paper, a new python library, PhysioEx, for physiological signal analysis through explainable deep learning is presented. The library offers a versatile and extensible API designed to provide developers with a standardized framework for sleep stage analysis. The API consists of 4 modules: `PhysioEx.preprocess`, `PhysioEx.data`, `PhysioEx.train`, and `PhysioEx.explain`. Users can easily add new or custom datasets into the library, analyze them with custom or state-of-the-art models, evaluate the generalization capability of such models across different datasets and compare them with other architectures trained in the same setup, and finally exploit new explainable AI methodologies to get insights into the models decision-making system.

The API has been built and tested to enable the usage of deep learning models on both low-resources machines and HPC clusters, such as torque and slurm, and proved to serve its purpose on servers equipped with NVIDIA T4 GPU and 24 GB RAM, NVDIA A30 GPU and 110 GB RAM, and on HPC clusters with each node equipped with 4 A100 GPUS and 512 GB RAM.

PhysioEx also implements a CLI toolbox to fasten the user development and deployment of deep learning models and the integration of new custom datasets. The CLI consists of 4 commands: `preprocess`, `train`, `test_model` and `finetune`. All the commands available on PhysioEx support the integration with `.yaml` config files to allow the users to integrate their custom datasets and models into the library. These commands have been tested in the production of the results presented in Tab 3 and Tab. 4.

Through the library API, the user can access different pretrained models on the Sleep Heart Health Study (SHHS) dataset which is one of the biggest available datasets for sleep staging[4, 7], consisting in recordings acquired from 5.463 subjects. The pretrained models available into PhysioEx comes with 2 different setups: one single channel EEG setup, and one multichannel EEG EOG and EMG. Thanks to this interface, users can easily fine-tune existing pre-trained networks with a considerably lower amount of training-epochs on their custom datasets, saving time and reducing the costs (e.g, renting the GPUs nodes): in our analysis, we trained the models on SHHS for 100 epochs and fine-tuned for 5 epochs, surpassing the results of the trained-from-scratch experiments provided by the authors in Tab 2.

Via the `PhysioEx.explain` module API, users can access new explainable AI methodologies suited for physiological signal analysis to inspect the neural networks decision-making system. PhysioEx addresses the challenge posed to state-of-the-art methodologies for explainable deep learning in time-series analysis, such as attribution methods[24, 38, 39], e.g. Integrated Gradients or Expected Gradients, by analyzing key aspects of the model decision-making system resembling the clinical expertise on sleep stages. For instance, the difficulties of state-of-the-art attribution methods in providing useful explanations for the network proposed by Chambon et al. in [36] has been shown in Fig 5 and Fig 6. PhysioEx employs a new attribution method, i.e., Spectral Gradients, which extracts the most important spectral components for a neural network in the classification of a time-series provided as input. This approach proved to correctly extract Delta bands in the classification of NREM3 (Deep Sleep) sleep stages, and Beta bands in the classification of Wake stages, Fig 5 and Fig 6.

PhysioEx investigates alternative approaches to explain the decision-making system of a neural network for sleep staging, such as concepts based architectures and explanations. These systems are largely underexplored in autonomous sleep staging due to the absence of concepts labelled datasets. PhysioEx implements new methodologies of autonomous concepts learning to allow the labelling with conceptual information of existing benchmark datasets. Fig 7 shows the capability of such approaches to correctly identify as concepts the presence of Alpha waves in Wake stages and Delta waves in NREM3 stages.

PhysioEx has been developed to address a critical challenge in autonomous sleep staging: gaining trust between clinicians and the AI. The library makes a step forward in this direction by introducing a standard testing environment where it's possible to fairly test custom AI models against the state-of-the-art with different benchmark sleep staging datasets and by introducing new explainability tools, tailored to physiological signal analysis, to investigate why different models share different performances. The library has demonstrated its ability to align the model's explanations with clinical knowledge by correctly identifying relevant spectral components and autonomously learning meaningful concepts from sleep staging data.

As future work, PhysioEx will focus its efforts on the integration of new large-scale benchmark datasets for sleep staging, such as the Wisconsin Sleep Cohort (WSC). By expanding the number of available datasets, PhysioEx aims to incorporate multi-dataset training technologies, including generative models, to address the challenges of bias and low generalizability often encountered when training on single datasets with specific experimental conditions. To this end, future developments will include the integration of both wearable-based datasets and clinical datasets obtained from invasive setups, such as intracranial EEG recordings.

Furthermore, PhysioEx will provide support for datasets with a customizable number of scorers for each sleep epoch, such as the DREEM dataset, which includes five different scorers for the same patients. In this way, PhysioEx aims to address the problem of inter-scorer variability, and hence the low generalizability of the results provided in Table 4, by allowing users to train and test their approach against different scorers, and to develop methodologies capable of solving this limitation.

At the same time, the focus will be on making new explainability methods accessible for models trained across multiple datasets, ensuring a more robust and transparent interpretation of sleep staging models in diverse experimental and clinical environments.

Data Availability Statement

The following datasets used in this study are publicly available for research purposes:

- **Sleep Heart Health Study (SHHS):** The SHHS dataset can be accessed through the National Sleep Research Resource (NSRR) at `https://sleepdata.org/datasets/shhs` under its data use agreement.
- **Montreal Archive of Sleep Studies (MASS):** The MASS dataset is available for academic research purposes at `https://borealisdata.ca/dataverse/MASS` under request.
- **Multi-Ethnic Study of Atherosclerosis (MESA):** The MESA Sleep dataset is accessible through the NSRR platform at `https://sleepdata.org/datasets/mesa` after completing the required data use agreements.
- **Outcomes of Sleep Disorders in Older Men (MROS):** The MROS dataset can be obtained from the NSRR at `https://sleepdata.org/datasets/mros`, subject to data use agreements.
- **Haaglanden Medisch Centrum (HMC):** The HMC dataset is available for research in the PhysioNet archive `https://doi.org/10.13026/t4w7-3k21`.
- **Dream Challenge Sleep Medicine (DCSM):** The DCSM dataset is publicly accessible via the DREAM Challenges platform at `https://doi.org/10.17894/ucph.282d3c1e-9b98-4c1e-886e-704afdfa9179`.

Acknoledgements

Financial support was provided by the Research Foundation Flanders(FWO) via SBO mandate 1SH4Z24N. FWO SBO Project "Supporting the development of self-regulation in infants: a promising strategy in preventive mental health care", S003524N. FWO Research Projects: 'Artificial Intelligence(AI) for data-driven personalized medicine', G0C9623N; 'Deep, personalized epileptic seizure detection', G0D8321N.

The National Heart, Lung, and Blood Institute provided funding for the ancillary MrOS Sleep Study, "Outcomes of Sleep Disorders in Older Men," under the following grant numbers: R01 HL071194, R01 HL070848, R01 HL070847, R01 HL070842, R01 HL070841, R01 HL070837, R01 HL070838, and R01 HL070839.

The Multi-Ethnic Study of Atherosclerosis (MESA) Sleep Ancillary study was funded by NIH-NHLBI Association of Sleep Disorders with Cardiovascular Health Across

Ethnic Groups (RO1 HL098433). MESA is supported by NHLBI funded contracts HHSN268201500003I, N01-HC-95159, N01-HC-95160, N01-HC-95161, N01-HC-95162, N01-HC-95163, N01-HC-95164, N01-HC-95165, N01-HC-95166, N01-HC-95167, N01-HC-95168 and N01-HC-95169 from the National Heart, Lung, and Blood Institute, and by cooperative agreements UL1-TR-000040, UL1-TR-001079, and UL1-TR-001420 funded by NCATS.

The Sleep Heart Health Study (SHHS) was supported by National Heart, Lung, and Blood Institute cooperative agreements U01HL53916 (University of California, Davis), U01HL53931 (New York University), U01HL53934 (University of Minnesota), U01HL53937 and U01HL64360 (Johns Hopkins University), U01HL53938 (University of Arizona), U01HL53940 (University of Washington), U01HL53941 (Boston University), and U01HL63463 (Case Western Reserve University).

The National Sleep Research Resource was supported by the National Heart, Lung, and Blood Institute (R24 HL114473, 75N92019R002).

Work partially supported by the Italian Ministry of Education and Research (MUR) in the framework of the FoReLab project (Departments of Excellence), and by the European Commission under the NextGenerationEU program, Partenariato Esteso PNRR PE1 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", and by PNRR - M4 C2, Investment 1.5 "Creating and strengthening of "innovation ecosystems", building "territorial R&D leaders", project "THE - Tuscany Health Ecosystem", Spoke 6 "Precision Medicine and Personalized Healthcare".

We thank Matteo Manni, a master student at the University of Pisa, for providing the plots of the concept-learning system presented in this article as part of his Master Thesis.

References

[1] Navya Baranwal, K Yu Phoebe, and Noah S Siegel. “Sleep physiology, pathophysiology, and sleep hygiene”. In: *Progress in cardiovascular diseases* 77 (2023), pp. 59–69.

[2] Huy Phan and Kaare Mikkelsen. “Automatic sleep staging of EEG signals: recent development, challenges, and future directions”. In: *Physiological Measurement* 43.4 (2022), 04TR01.

[3] Ary L Goldberger et al. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals”. In: *circulation* 101.23 (2000), e215–e220.

[4] Stuart F Quan et al. “The sleep heart health study: design, rationale, and methods”. In: *Sleep* 20.12 (1997), pp. 1077–1085.

[5] Terri Blackwell et al. "Associations between sleep architecture and sleep-disordered breathing and cognition in older community-dwelling men: the osteoporotic fractures in men sleep study". In: *Journal of the American Geriatrics Society* 59.12 (2011), pp. 2217–2225.

[6] Xiaoli Chen et al. "Racial/ethnic differences in sleep disturbances: the Multi-Ethnic Study of Atherosclerosis (MESA)". In: *Sleep* 38.6 (2015), pp. 877–888.

[7] Guo-Qiang Zhang et al. "The National Sleep Research Resource: towards a sleep data commons". In: *Journal of the American Medical Informatics Association* 25.10 (2018), pp. 1351–1358.

[8] Seyedeh Neelufar Payrovnaziri et al. "Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review". In: *Journal of the American Medical Informatics Association* 27.7 (2020), pp. 1173–1185.

[9] Patrick Schramowski et al. "Making deep neural networks right for the right scientific reasons by interacting with their explanations". In: *Nature Machine Intelligence* 2.8 (2020), pp. 476–486.

[10] Madeleine M Grigg-Damberger. "The AASM Scoring Manual four years later". In: *Journal of Clinical Sleep Medicine* 8.3 (2012), pp. 323–332.

[11] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42 (2017), pp. 60–88.

[12] Antonio L Alfeo, Mario GCA Cimino, Guido Gagliardi, et al. "Improving an Ensemble of Neural Networks via a Novel Multi-class Decomposition Schema." In: *IJCCI*. 2022, pp. 389–395.

[13] Emina Alickovic and Abdulhamit Subasi. "Ensemble SVM method for automatic sleep stage classification". In: *IEEE Transactions on Instrumentation and Measurement* 67.6 (2018), pp. 1258–1265.

[14] Varun Bajaj and Ram Bilas Pachori. "Automatic classification of sleep stages based on the time-frequency image of EEG signals". In: *Computer methods and programs in biomedicine* 112.3 (2013), pp. 320–328.

[15] Antonio L Alfeo, Mario GCA Cimino, Guido Gagliardi, et al. "Automatic Feature Extraction for Bearings' Degradation Assessment using Minimally Pre-processed Time series and Multi-modal Feature Learning." In: *IN4PL*. 2022, pp. 94–103.

[16] Orestis Tsinalis et al. "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks". In: *arXiv preprint arXiv:1610.01683* (2016).

[17] Siddharth Biswal et al. "SLEEPNET: automated sleep staging system via deep learning". In: *arXiv preprint arXiv:1707.08262* (2017).

[18] Konstantinos Kontras et al. "CoRe-Sleep: A Multimodal Fusion Framework for Time Series Robust to Imperfect Modalities." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2024).

[19] E Heremans et al. "Automatic sleep staging and detection of sleep disorders through wearable EEG monitoring devices". In: *Sleep Medicine* 115 (2024), S231–S232.

[20] Huy Phan et al. "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.3 (2019), pp. 400–410.

[21] Antonio Luca Alfeo, Mario GCA Cimino, Guido Gagliardi, et al. "Matching the expert's knowledge via a counterfactual-based feature importance measure". In: *Proceedings of the 5th XKDD Workshop-European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2023)*. 2023.

[22] Guido Gagliardi et al. "Improving emotion recognition systems by exploiting the spatial information of EEG sensors". In: *IEEE Access* (2023), pp. 1–1. ISSN: 2169-3536. DOI: 10.1109/access.2023.3268233.

[23] Guido Gagliardi et al. "Fine-grained Emotion Recognition using Brain-Heart Interplay measurements and eXplainable Convolutional Neural Networks". In: *Proceedings of the 11th International IEEE EMBS Conference on Neural Engineering*. 2023, pp. 1–1.

[24] Andreas Theissler et al. "Explainable AI for time series classification: a review, taxonomy and research directions". In: *Ieee Access* 10 (2022), pp. 100700–100724.

[25] Mateo Espinosa Zarlenga et al. "Concept embedding models: Beyond the accuracy-explainability trade-off". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21400–21413.

[26] Guido Gagliardi et al. "Using contrastive learning to inject domain-knowledge into neural networks for recognizing emotions". In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2023, pp. 1587–1592.

[27] Antonio Luca Alfeo, Mario G. C. A. Cimino, and Guido Gagliardi. "Concept-wise granular computing for explainable artificial intelligence". In: *Granular Computing* (2022). ISSN: 2364-4966. DOI: 10.1007/s41066-022-00357-8.

[28] Antonio Luca Alfeo, Mario GCA Cimino, and Guido Gagliardi. "Recognizing Bearings' Degradation Stage Using Multimodal Autoencoder to Learn Features from Different Time Series". In: *SN Computer Science* 5.4 (2024), p. 371.

[29] Wolfgang Stammer et al. "Interactive Disentanglement: Learning Concepts by Interacting With Their Prototype Representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10317–10328.

[30] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[31] Narine Kokhlikyan et al. "Captum: A unified and generic model interpretability library for pytorch". In: *arXiv preprint arXiv:2009.07896* (2020).

[32] Christian O'reilly et al. "Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research". In: *Journal of sleep research* 23.6 (2014), pp. 628–635.

[33] Diego Alvarez-Estevez and Roselyne M Rijsman. "Inter-database validation of a deep learning approach for automatic sleep scoring". In: *PloS one* 16.8 (2021), e0256111.

[34] Mathias Perslev et al. "U-Sleep: resilient high-frequency sleep staging". In: *NPJ digital medicine* 4.1 (2021), p. 72.

[35] Huy Phan et al. "XSleepNet: Multi-view sequential model for automatic sleep staging". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), pp. 5903–5915.

[36] Stanislas Chambon et al. "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.4 (2018), pp. 758–769.

[37] Akara Supratak and Yike Guo. "TinySleepNet: An efficient deep learning model for sleep stage scoring based on raw single-channel EEG". In: *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2020, pp. 641–644.

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International conference on machine learning*. PMLR. 2017, pp. 3319–3328.

[39] Gabriel Erion et al. "Improving performance of deep learning models with axiomatic attribution priors and expected gradients". In: *Nature machine intelligence* 3.7 (2021), pp. 620–631.